University of Málaga

Health Engineering

Laboratory Task

Neural networks

Author

Alejandro Domínguez Recio

Course

Intelligent Systems

Teachers

Enrique Domínguez Merino

Jesús de Benito Picazo

Introduction

In this practice we going to evaluate three diferents dataset with the Multilayer Perceptron classifier. We will modify some parameters of this such as learning rate and hidden layers in order to see how these affect the algorithm and its results.To evaluate which dataset is more optimal with this classifier we going to describe the performance measures seen in class like confusion matrix, acurracy, precision, fallout, recall, F-measure and area under ROC curve. Also we going to detail the characteristics of the different datasets.

How are we going to do it?

To do the different studies we will have the following structure in each of then.

- **Dataset context**

In this part we going to write a little introduction of the dataset context commenting on data type and prediction target.

- **Number of classes**

We going to describe the number of classes as if it is binary o multiclass. To do that we going to observe the atributtes weka panel and we going to select the class attribute and depending on the number of values that it takes, we will determine if it is binary or multiclass.

- **Number of attributes**

For the number of attributes we going to inspect the weka attributes panel or open the file in text mode and inspect the characteristics.

- **Number of samples or instances**

To know the number of samples we can proceed as in the previous section by inspecting the weka panels or opening the document in text mode and seeing its characteristics.

➔ Configuration

HiddenLayers

The number of layers will depend on the characteristics of the function that we want to represent with our neural network. For cases in which we want to represent linear functions, probably with a single layer it will be enough although in theory with two layers the total of the functions could be represented practically. In our case we are using a multilayer perceptron allowing us to represent more complex functions. When looking for the ideal number of layers we can opt for several strategies such as random or systematic experimentation, orientation by similar networks already implemented or simply intuition by the characteristics of the dataset in question.

LearningRate

Assuming that neural networks are trained using the stochastic descending gradient algorithm. This estimates the gradient error between the current state of the model and the result, which is used to update the model weights using backpropagation of the errors. The learning ratio, a value between 0 and 1 marks the amount of update of that error in the weights. A ratio of 0 marks a null learning from this error and a learning of 1 marks the total. We emphasize that a learning of 1, although it may seem a priori the most optimal, this creates an unbalanced learning by assigning the total learning to the model used in question, ignoring the rest of the options.

Momentum

Taking into account that the networks use the descending gradient algorithm to minimize the error in reaching a global minimum or what is the same, the parameters where our model presents greater accuracy. The use of the moment (value between 0 and 1) allows that in the search for the global minimum the algorithm does not get stuck in local minima of our possible function. Values close to 0 mark that our steps towards the minimum of our function are null and, on the contrary, values close to 1 mark large steps. We emphasize that this value is added to the product of the learning rate * weightgradient.

- ***Performance metric values by configuration***

Accuracy

We can obtain the acurracy in two ways, one of them is directly from the classifier out and the another one is by calculating the number of correctly predicted examples divided by the total number of examples.

Precision

This measurement shows the positive predictive value, higher is bettter. We are going to obtain the necessary values from the confusion matrix and perform the following calculation TP/(TP+FP).

Fallout

This measurement shows the false positive rate, lower is better. We are going to obtain the necessary values from the confusion matrix and perform the following calculation FP/(FP+TN).

Recall

This measurement shows the true positive rate, higher is better. We are going to obtain the necessary values from the confusion matrix and perform the

following calculation TP/(TP+FN).

F-measure

This measurement provides a single score that balances both the concers of precision and recall in one number, higher is better. We are going to obtain the necessary values of the previous measurements, precision and recall, and perform the following calculation 2*(Precision*Recall)/(Precision+Recall).

- **ROC curve and the area under the curve**

We are going to obtain this measurement by visualizing the threshold curve in Weka. The area under the ROC curve is a number in the interval [0,1], a higher value is better. This measure show the trade-off between the ratios of false positives and false negatives. This measurement is very useful when we faced with unbalanced data.

*The previous points will be repeated for each dataset

- **Evaluation of the results**

In this section we are goint to compare the results of the different performance measures.

- **Differences between datasets**

Here we are going to describe the main differences in the data of the different datasets.

- **Last conclusions**

Finally, we are going explain possible reasons why some datasets perform better with the Naive Bayes classifier than others. We will support our conclusions on the performance measures taken and on the characteristics of the datasets

BREAST CANCER DATASET

➔ Introduction

The main idea of this classification problem is given a dataset about patients with a series of characteristics which determine if they have recurrence events or no recurrence events of breast cancer train a Multilayer Perceptron classification model and evaluate its performance.

In this dataset we have 286 instances in total of which 201 of one class and 85 instances of another class. Each instance has ten attributes, one of them is the class attribute. Because of we have only two classes we are faced with a binary classification problem.

➔ Number of classes

In this case our class attribute can take only **two *values***, *recurrence events* and *no recurrence events*. Because of that we are faced to a ***binary classification problem***.

➔ Number of attributes

In this case we have **ten attributes** which are *class, age, menopause, tumor-size, inv-nodes, node-caps, deg-malig, breast, breast-quad* and *irradiat.*

➔ Number of samples

In this case in particular the number of samples or instances is **286.**

- **Performance metric values by configuration**

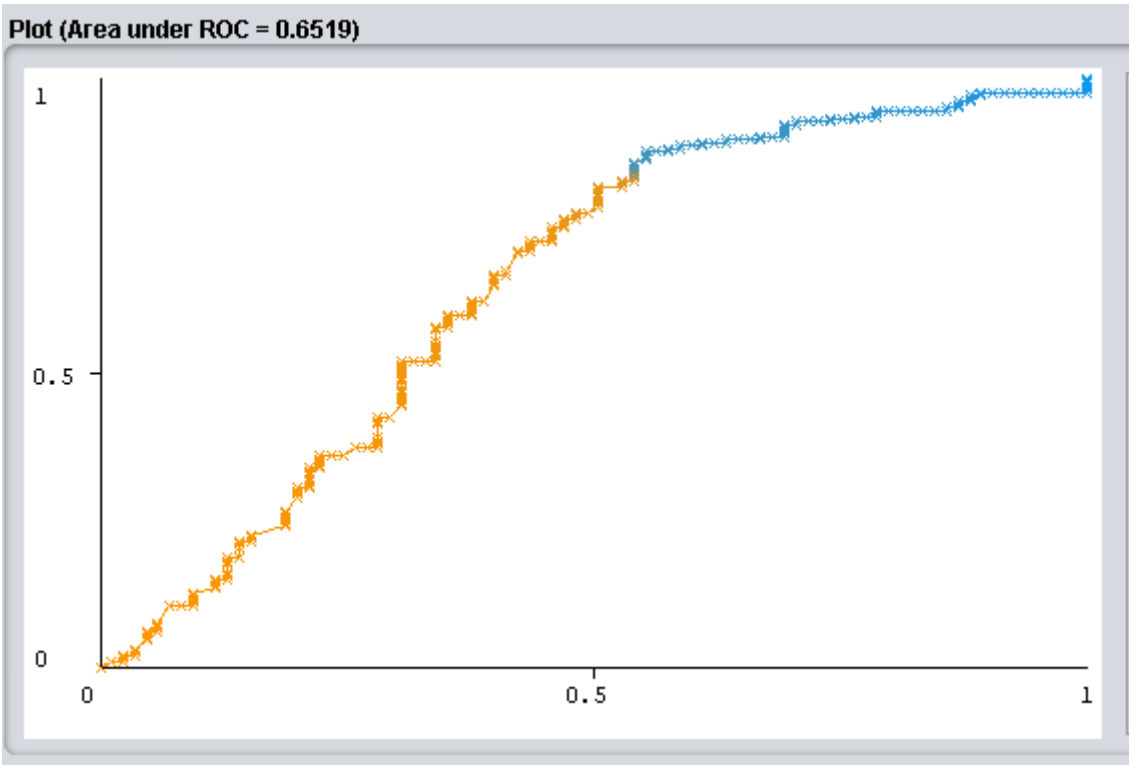
Configurations

Configuration	HiddenLayers	LearningRate	Momentum
C1	1	0,3	0,3
C2	2	0,15	0,15
C3	1	0,01	0,01
C4	1	0,7	0,7
C5	2	0,7	0,7
C6	2	0,01	0,01
C7	2	0,01	0,7
C8	1	0,01	0,7

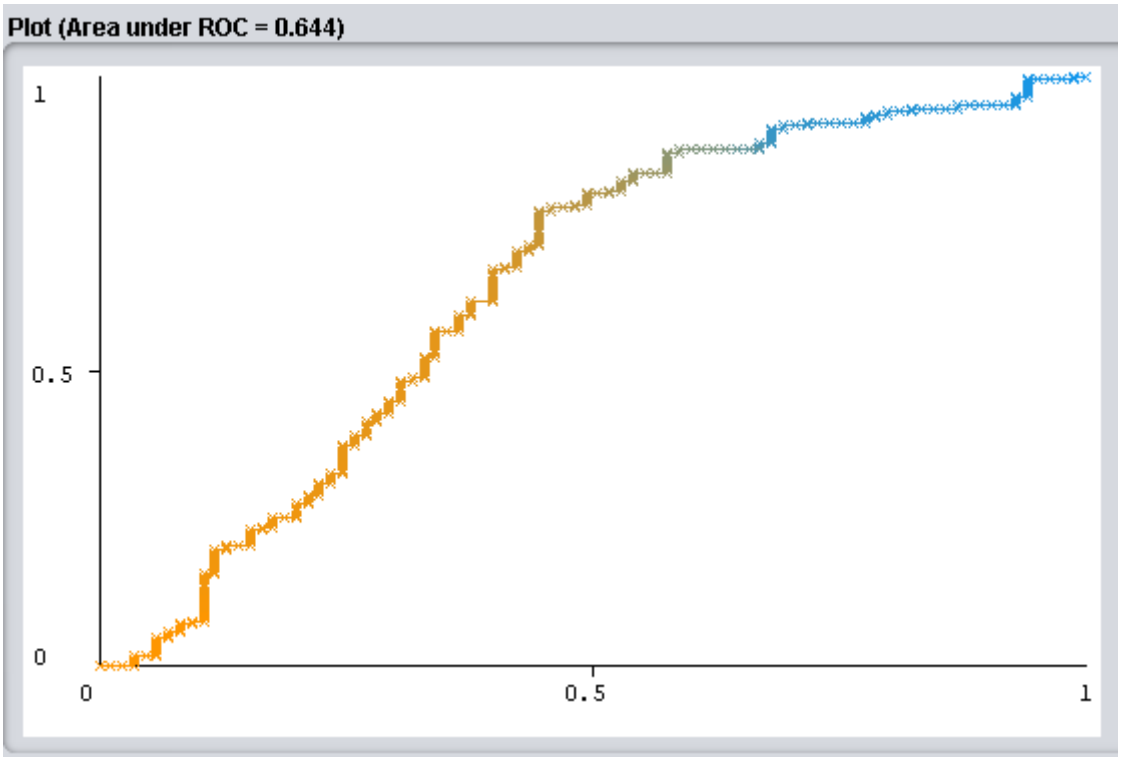
Results

Dataset	Acurracy	Precision	Fallout	Recall	F-measure	ROC	Time
C1	72,028	0,784	0,541	0,831	0,807	0,6519	0,23
C2	73,426	0,778	0,588	0,871	0,822	0,644	0,37
C3	70,979	0,771	0,588	0,836	0,802	0,667	0,23
C4	71,328	0,772	0,588	0,841	0,805	0,652	0,22
C5	72,7273	0,776	0,588	0,861	0,816	0,690	0,35
C6	74,4755	0,783	0,576	0,881	0,829	0,678	0,35
C7	74,8252	0,787	0,565	0,881	0,831	0,664	0,39
C8	70,6294	0,772	0,576	0,826	0,798	0,642	0,22

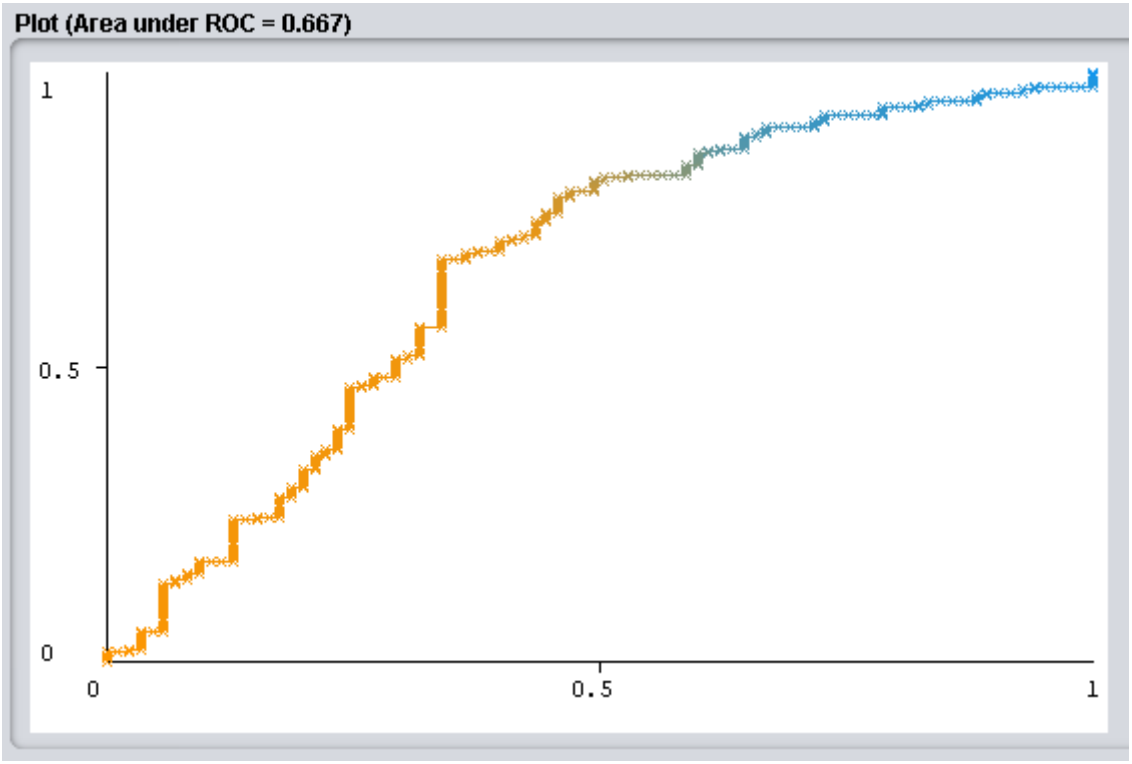
➔ ROC curve C1



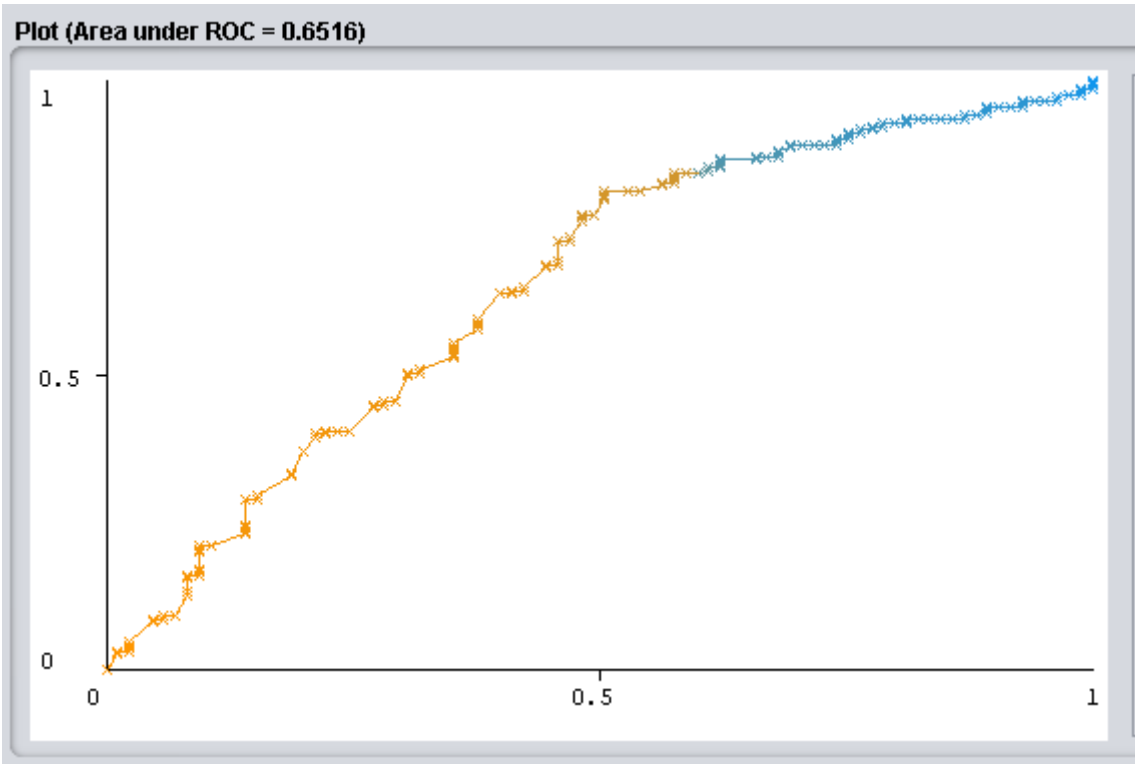
➔ROC curve C2



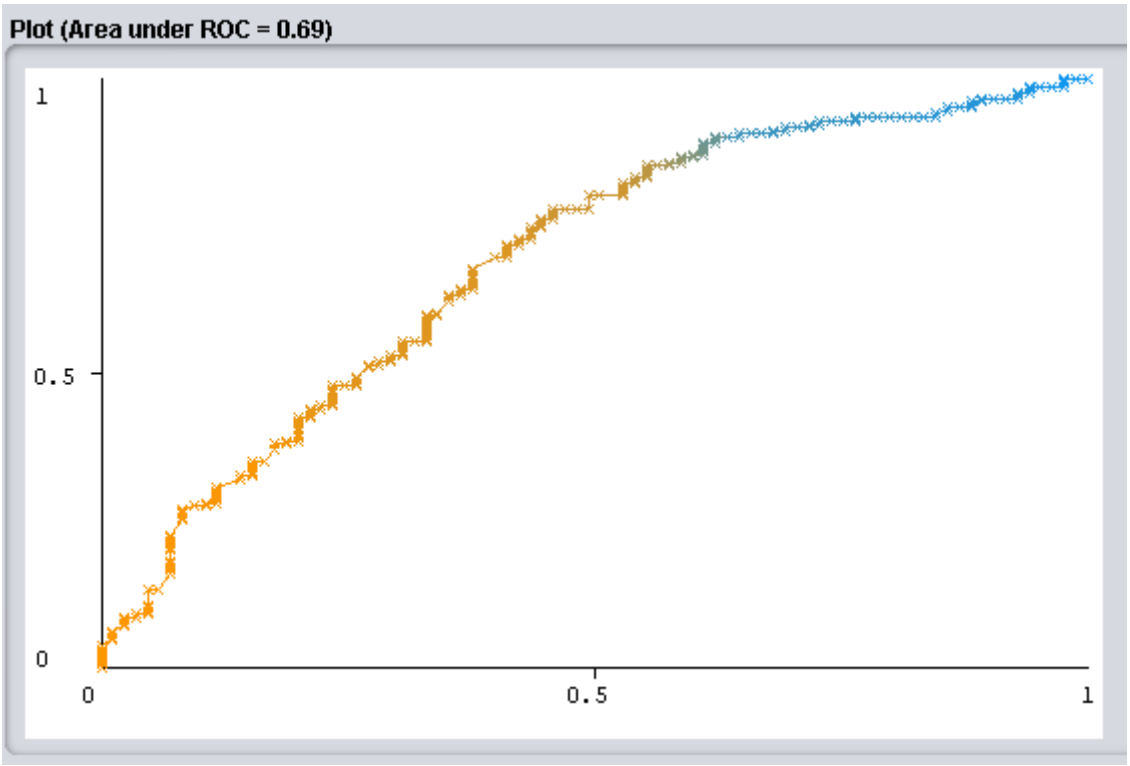
➔ROC curve C3



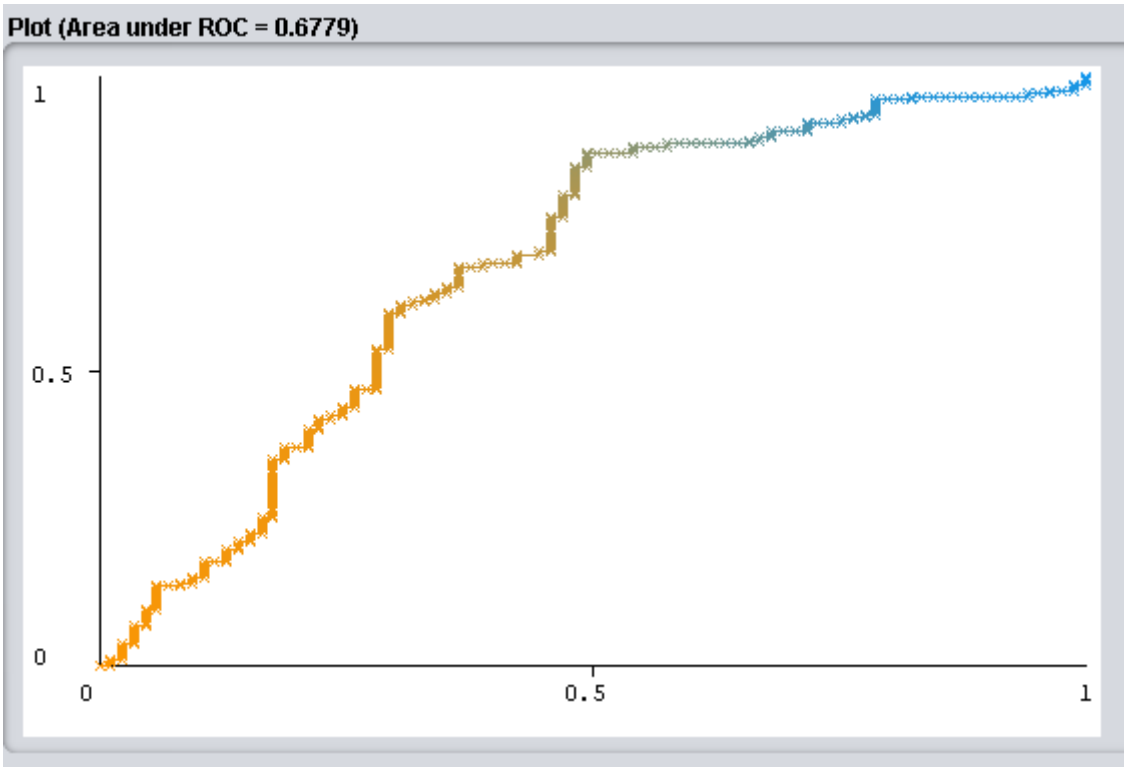
➔ROC curve C4



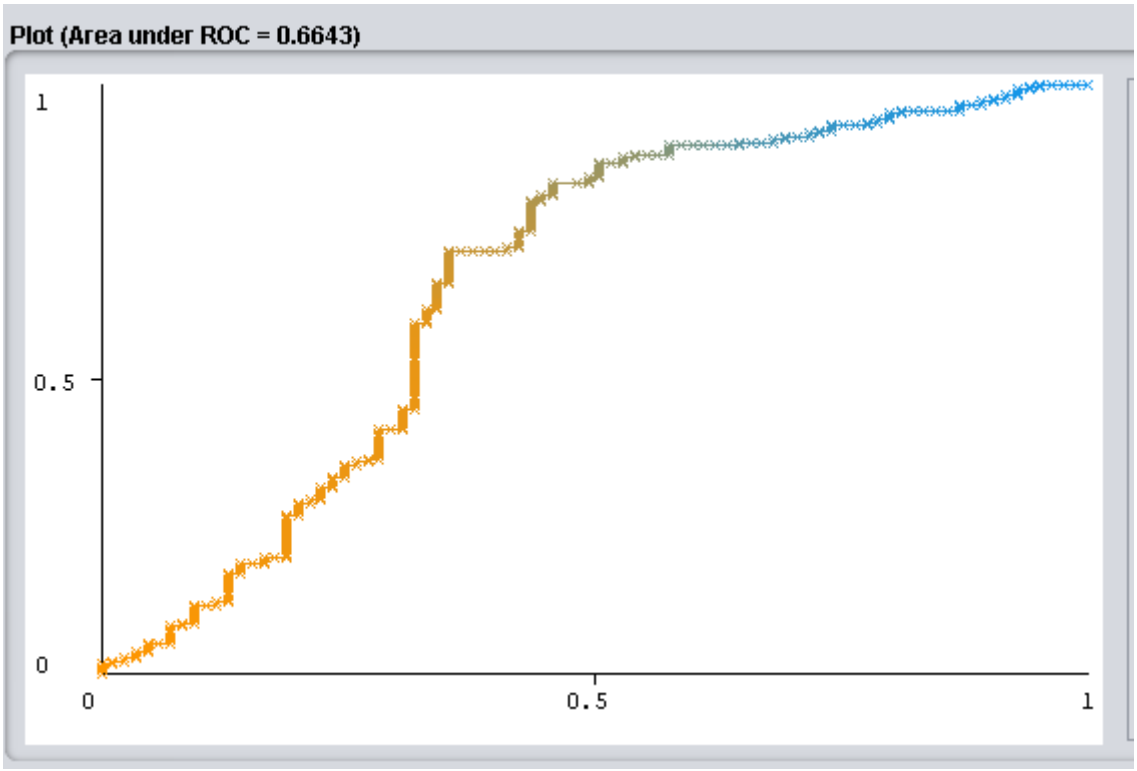
➔ROC curve C5



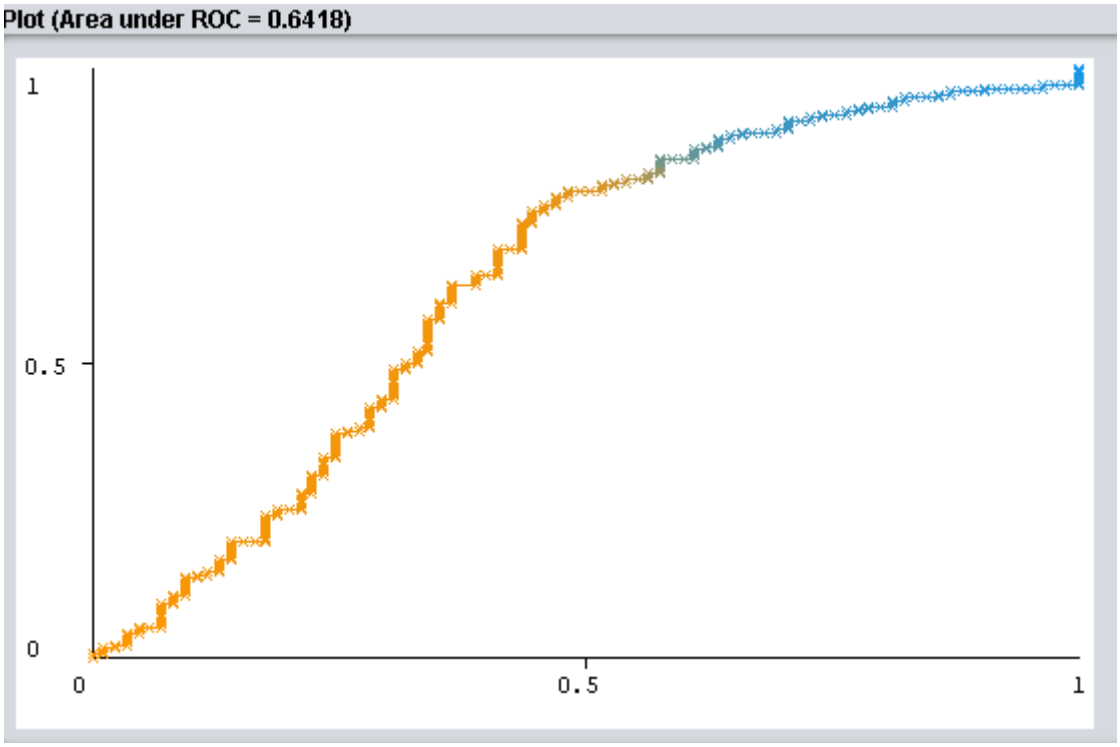
➔ROC curve C6



➔ROC curve C7



➔ROC curve C8



➔ Section conclusions

Among the main differences that we can find is that the tests in which we have used 1 layer have had a slower execution time, although in the C1 the results of the performance metrics have been very similar to the best with a difference of just a few. tenths. Speaking of the 2-layer combinations we can say that they have given the best results globally. Finally I add that no differences are observed when we use extreme values in the learning rate or momentum, so we could say that the model finds the 'global' minimum of the gradient well or this being a possible sign that our function does not have many beyond of the finally obtained.

HEPATITIS DATASET

➔ Introduction

The main idea of this classification problem is given a dataset about patients which suffer from hepatitis and they a series of characteristics which determine if they died or live because train a Multilayer Perceptron model and evaluate its performance.

In this dataset we have 155 instances in total of which 129 of one class and 26 instances of another class. Each instance has 20 attributes, one of them is the class attribute. Because of we have only two classes we are faced with a binary classification problem.

➔ Number of classes

In this case our class attribute can take only ***two values***, *die* and *live*. Because of that we are faced to a ***binary classification problem***.

➔ Number of attributes

In this case we have **twenty attributes** which are *age, sex, steroid, antivirals, fatigue, malaise, anorexia, liver big, liver firm, spleen palpable, spiders, ascites, varices, bilirubin, alk phosphate, sgot, albumin, protime, histology* and *class.*

➔ Number of samples

In this case in particular the number of samples or instances is **155.**

➔ **Performance metrics values by configuration**

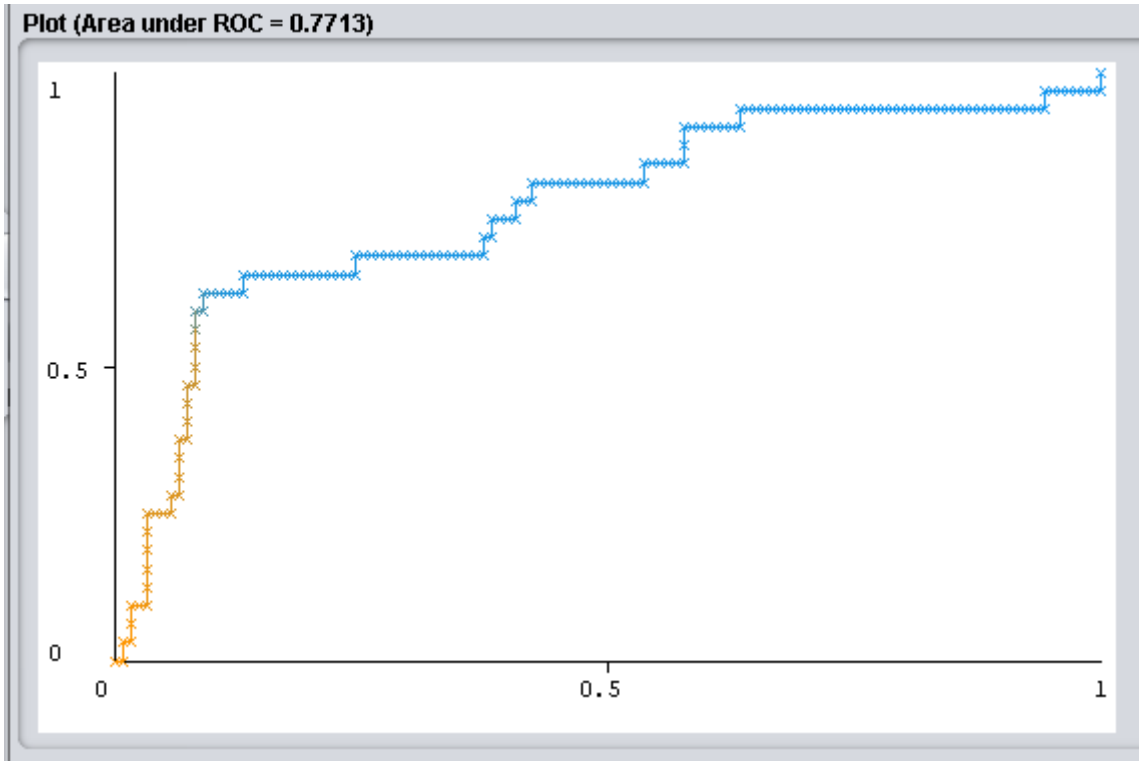
Configurations

Configuration	HiddenLayers	LearningRate	Momentum
C1	1	0,3	0,3
C2	2	0,15	0,15
C3	1	0,01	0,01
C4	1	0,7	0,7
C5	2	0,7	0,7
C6	2	0,01	0,01
C7	2	0,01	0,7
C8	1	0,01	0,7

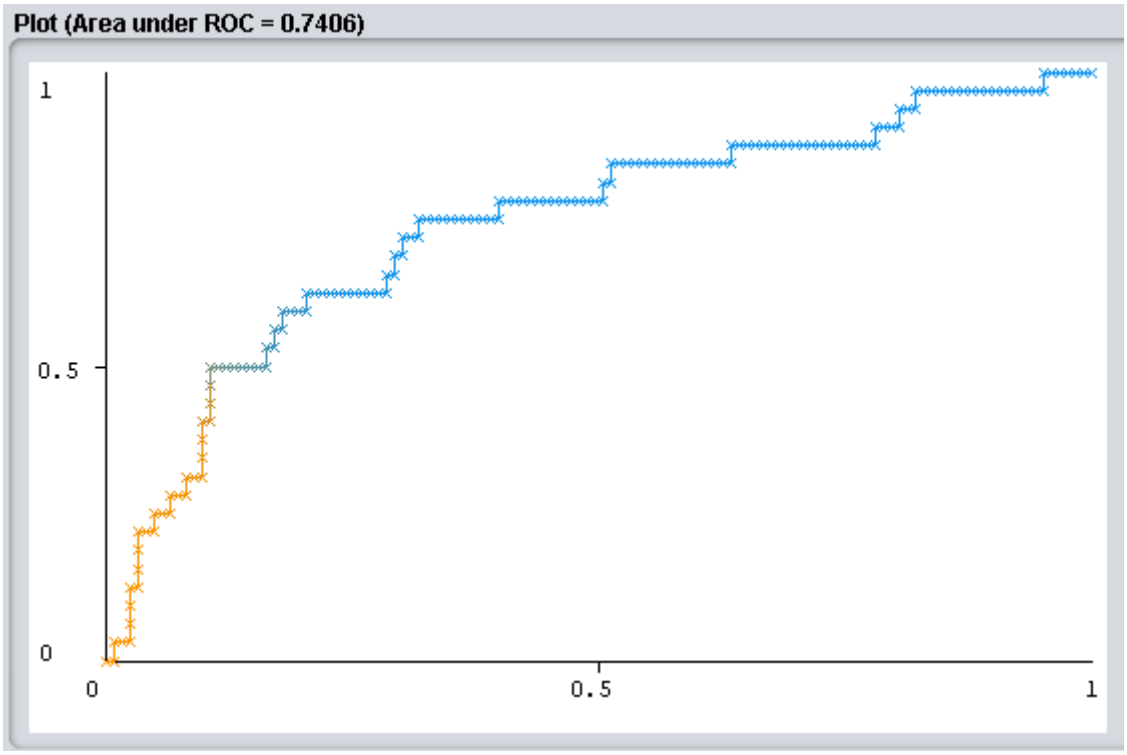
Results

Dataset	Acurracy	Precision	Fallout	Recall	F-measure	ROC	Time
C1	83,871	0,630	0,081	0,531	0,576	0,771	0,07
C2	80,6452	0,536	0,105	0,469	0,500	0,741	0,1
C3	83,871	0,613	0,098	0,594	0,603	0,799	0,06
C4	83,871	0,606	0,106	0,625	0,615	0,795	0,06
C5	85,1613	0,655	0,081	0,594	0,623	0,765	0,1
C6	84,5161	0,633	0,089	0,594	0,613	0,807	0,09
C7	81,2903	0,545	0,122	0,563	0,554	0,782	0,09
C8	81,2903	0,543	0,130	0,594	0,567	0,796	0,06

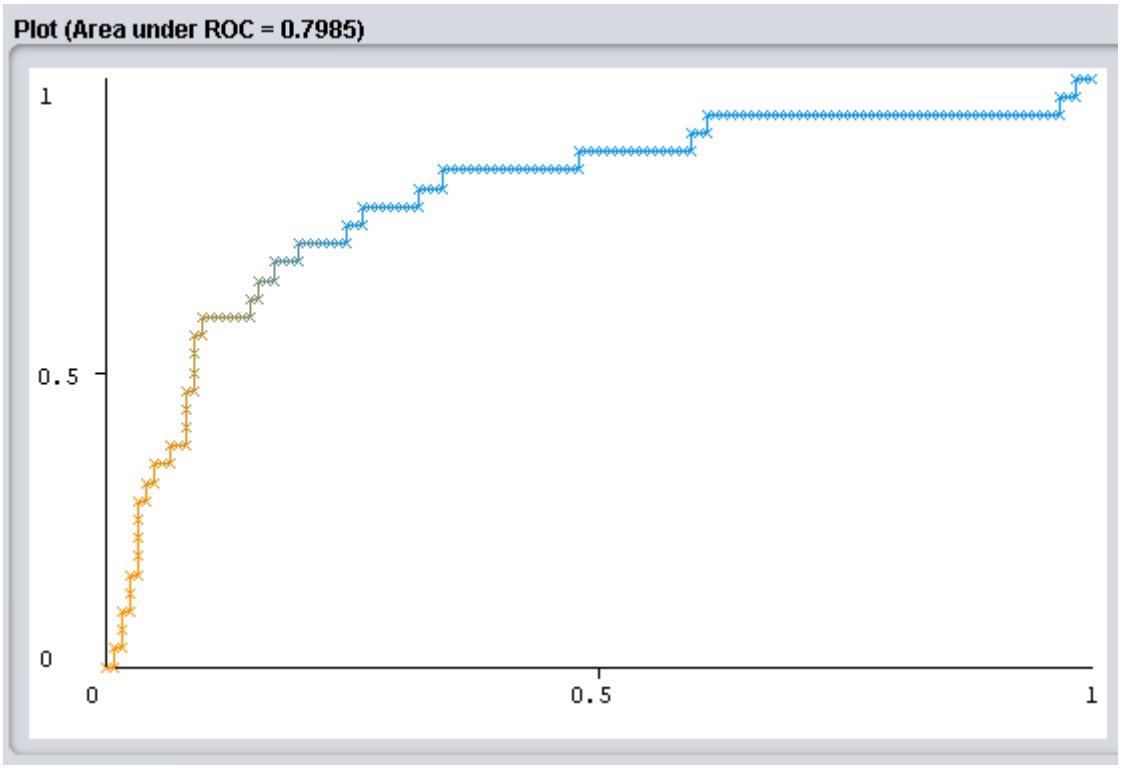
➔ROC curve C1



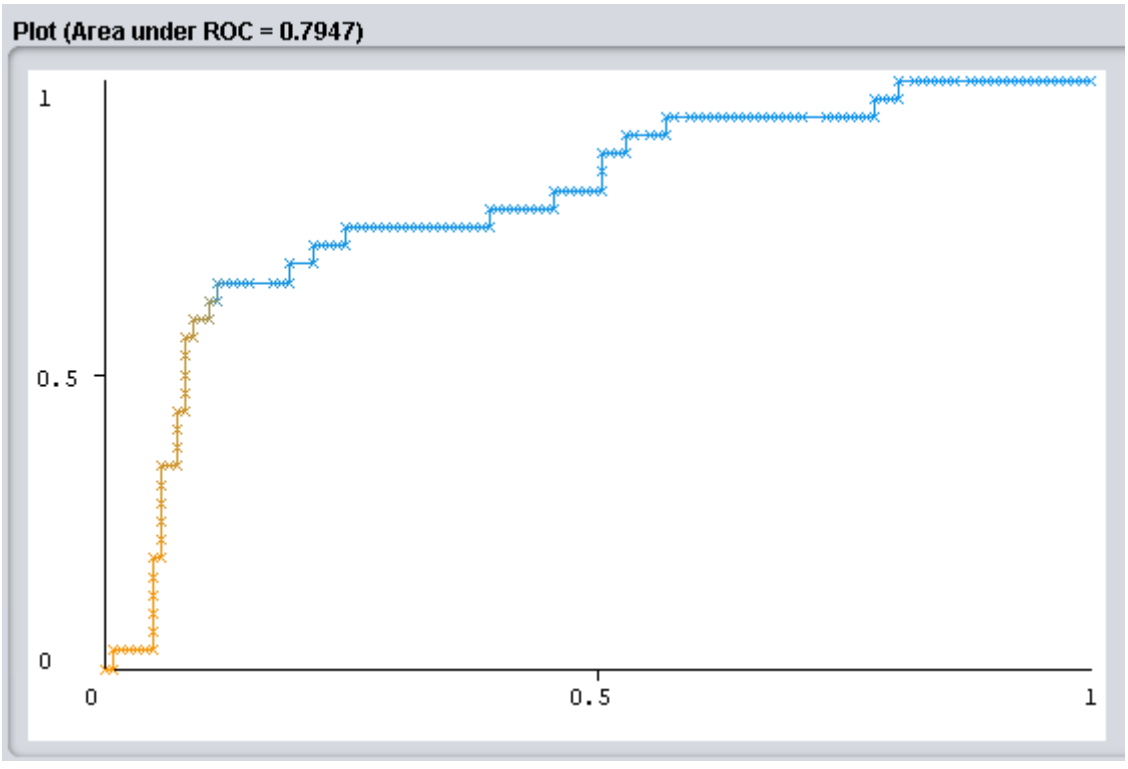
➔ROC curve C2



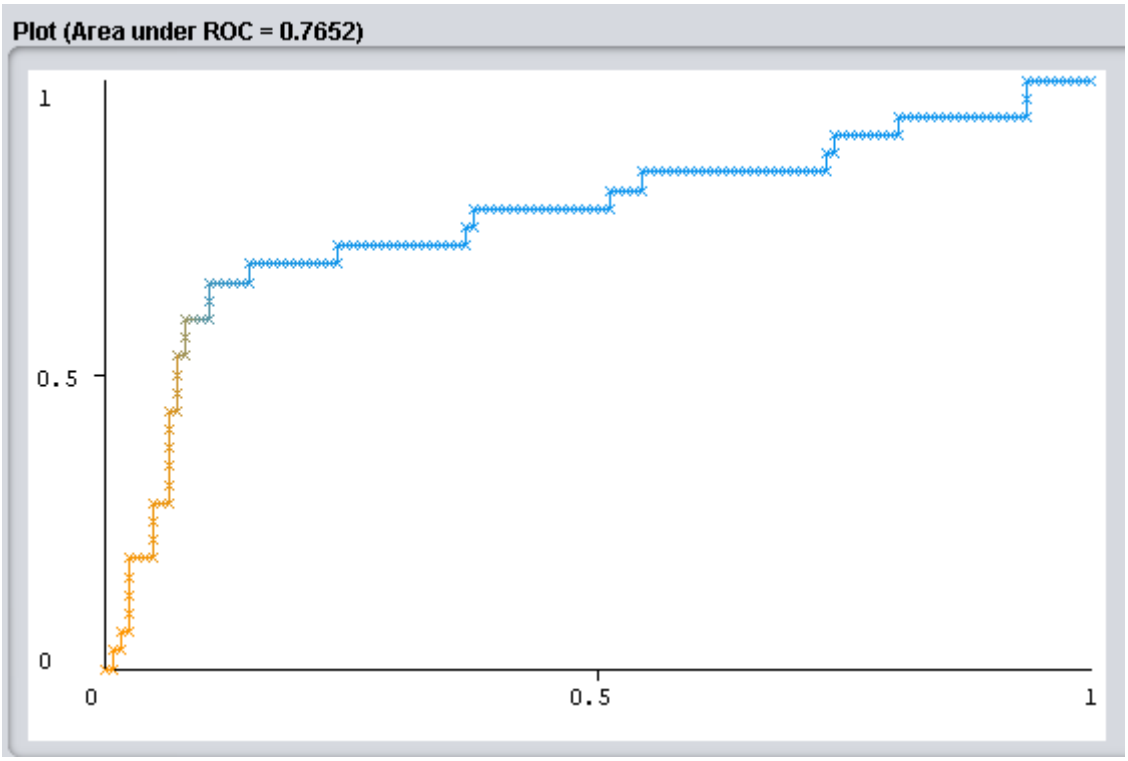
➔ROC curve C3



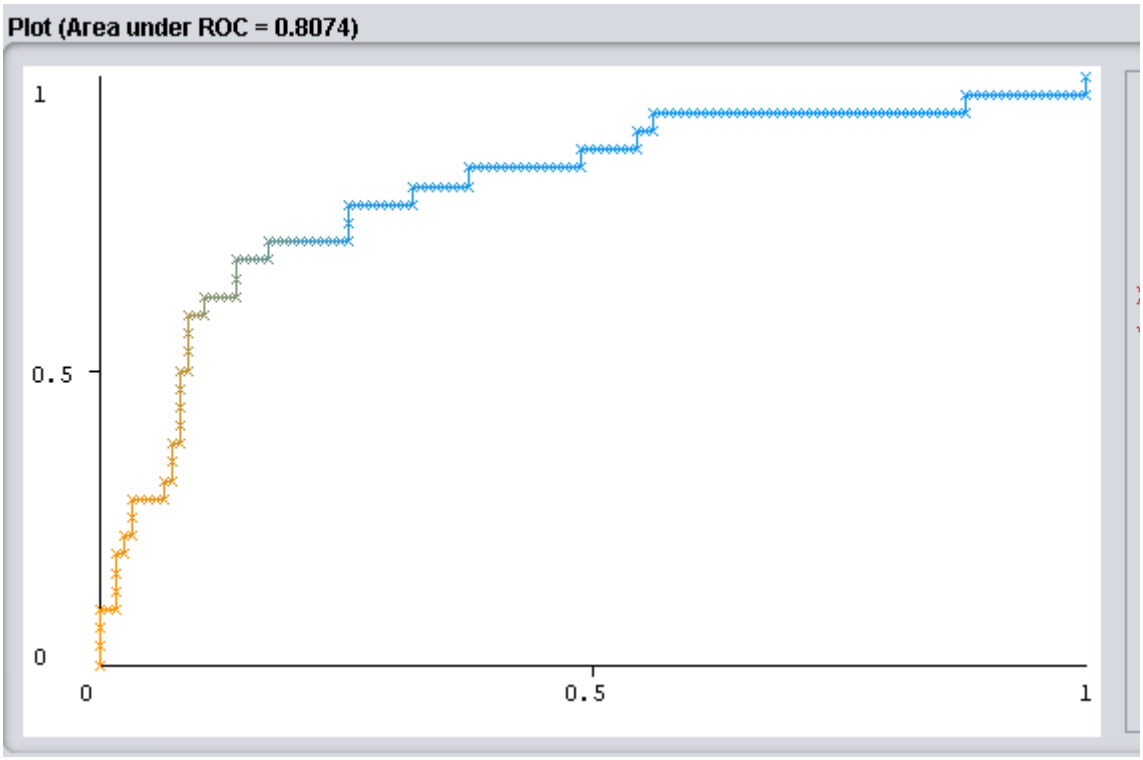
➔ROC curve C4



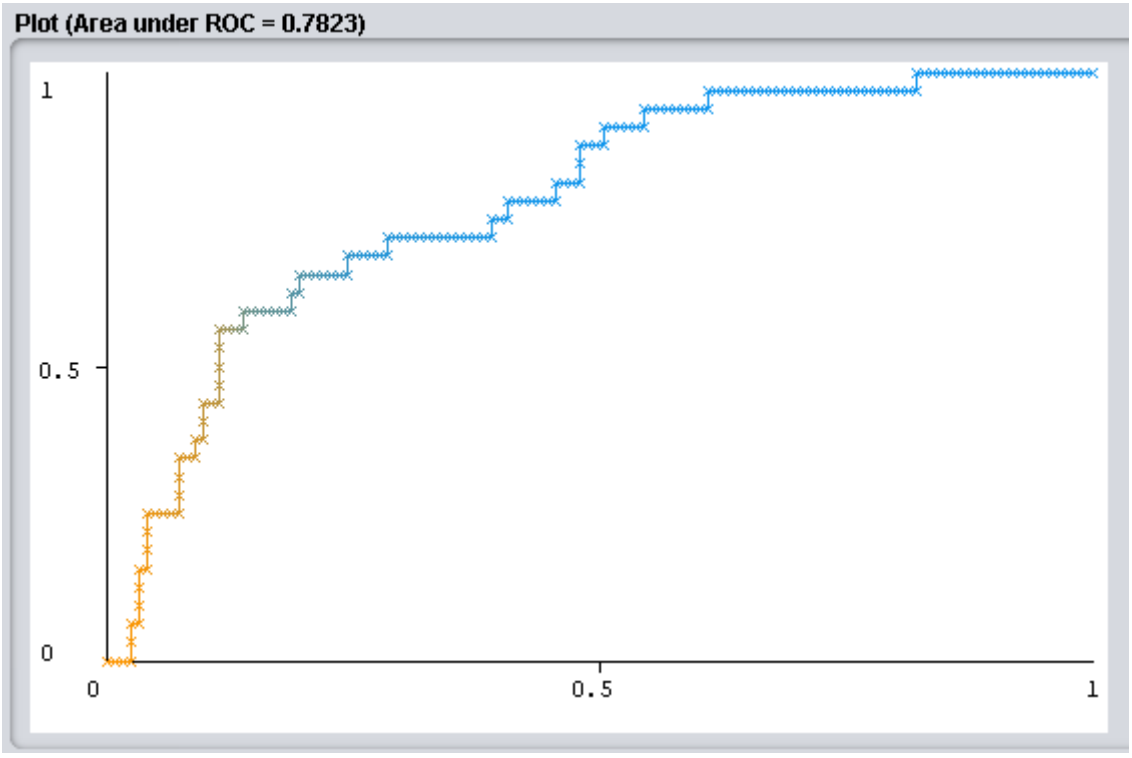
➔ROC curve C5



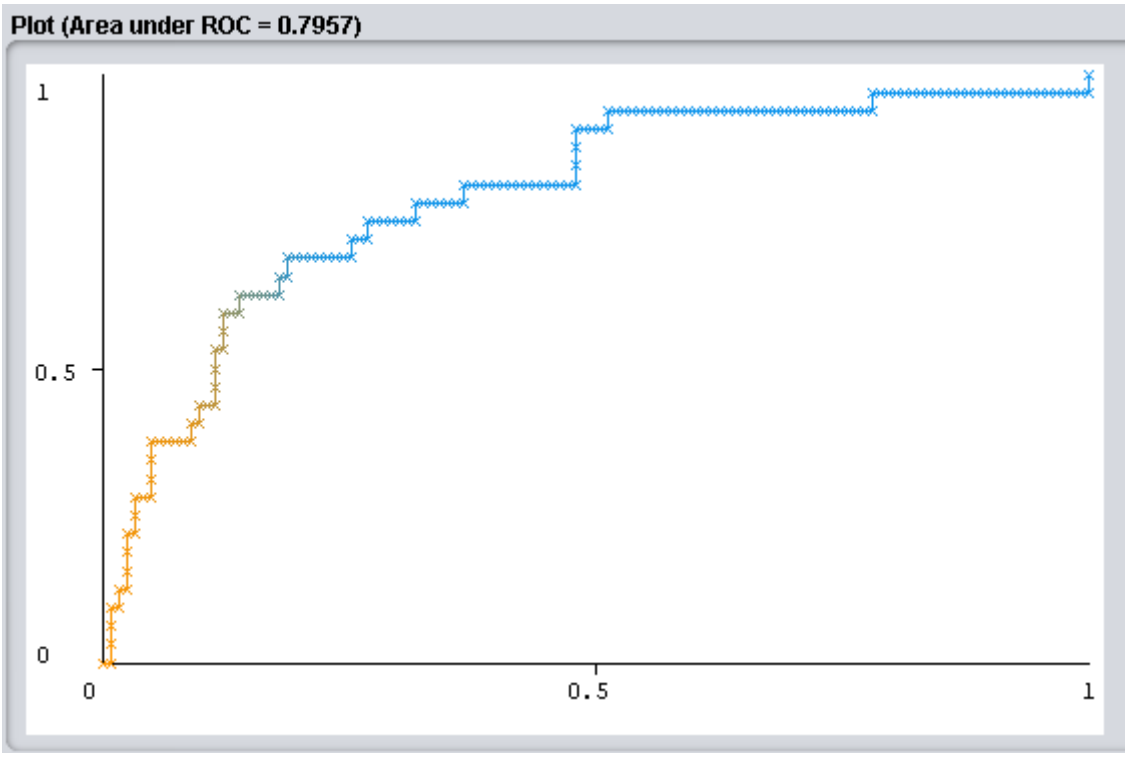
➔ROC curve C6



➔ROC curve C7



➔ROC curve C8



➔ Section conclusions

First of all we can observe a difference in execution time between the 1 and 2 layer configurations as expected. On the other hand, we have that the differences in learning rate or momentum between the configurations with the same number of layers do not cause a significant effect, which gives us a possible signal that our functions do not have local minima in which the gradient vector can be 'trapped. '. If we talk about which configuration has the best result, we could say that C6 shows, coinciding with the fact that it has 2 hidden layers and a learning rate and low momentum of 0.01. The difference in the result with C5 may be because it has a higher learning rate, this step size causes the gradient to pass above minimums more easily, thus losing optimal configurations.

POST-OPERATIVE PATIENTS DATASET

➔ Introduction

The main idea of this classification problem is given a dataset about patients which have passed an operation and they are in a postoperative recovery area waiting to a decision of where they should be sent next. Depending on a number of temperature measurements we train a Multilayer Perceptron model and evaluate its performance.

In this dataset we have 90 instances in total of which they are divided in three categories or classes, 2 of class I, 24 of class S and 64 of class A. Each instance has 9 attributes, one of them is the class attribute. Because of we have three possible classes we are faced with a multiclass classification problem.

➔ Number of classes

In this case our class attribute can take only **three *values***, I, S and *A*. Because of that we are faced to a **multiclass *classification problem***.

➔ Number of attributes

In this case we have **nine attributes** which *l-core, l-surf, l-02, l-bp, surf-stbl, core-stbl, bp-stbl, comfort* and *class.*

➔ Number of samples

In this case in particular the number of samples or instances is **90.**

➔ Performance metrics values by configuration

Configurations

Configuration	HiddenLayers	LearningRate	Momentum
C1	1	0,3	0,3
C2	2	0,15	0,15
C3	1	0,01	0,01
C4	1	0,7	0,7
C5	2	0,7	0,7
C6	2	0,01	0,01
C7	2	0,01	0,7
C8	1	0,01	0,7

Results

Dataset	Acurracy	Precision	Fallout	Recall	F-measure	ROC	Time
C1	60	?	0,705	0,6	?	0,403	0,06
C2	63,333	?	0,646	0,633	?	0,368	0,07
C3	71,111	?	0,711	0,711	?	0,328	0,04
C4	66,666	?	0,681	0,667	?	0,447	0,05
C5	61,111	?	0,608	0,611	?	0,468	0,07
C6	68,888	?	0,719	0,689	?	0,328	0,06
C7	61,111	?	0,701	0,611	?	0,323	0,07
C8	60	?	0,682	0,6	?	0,328	0,04

➔ Section conclusions

Regarding the execution times, in the first place we can say that as it has been repeated in the previous data sets, those configurations with 2 hidden layers have required more time, although we are surprised by the value of C1 with 0.6, being a time equal to configurations with 2 layers like C6, although taking into account that the latter has a low learning rate of 0.01, so the time of the gradient vector to reach the minimum is greater due to the size of its step, affecting the execution time. About which configuration gives the best ROC result, we can say that it is C5, highlighting that this is a configuration with a learning rate and high momentum of 0.7 this may be because our function has a minimum clear of the opposite with this step size it would be relatively easy get out of this if it were local, however they are not very effective measures.

➢ **Evaluation of the results**

Dataset	Acurracy	Precision	Fallout	Recall	F-measure	ROC
BreastCancerC1	72,028	0,784	0,541	0,831	0,807	0,6519
BreastCancerC2	73,426	0,778	0,588	0,871	0,822	0,644
BreastCancerC3	70,979	0,771	0,588	0,836	0,802	0,667
BreastCancerC4	71,328	0,772	0,588	0,841	0,805	0,652
BreastCancerC5	72,7273	0,776	0,588	0,861	0,816	0,690
BreastCancerC6	74,4755	0,783	0,576	0,881	0,829	0,678
BreastCancerC7	74,8252	0,787	0,565	0,881	0,831	0,664
BreastCancerC8	70,6294	0,772	0,576	0,826	0,798	0,642
HepatitisC1	83,871	0,630	0,081	0,531	0,576	0,771
HepatitisC2	80,6452	0,536	0,105	0,469	0,500	0,741
HepatitisC3	83,871	0,613	0,098	0,594	0,603	0,799
HepatitisC4	83,871	0,606	0,106	0,625	0,615	0,795
HepatitisC5	85,1613	0,655	0,081	0,594	0,623	0,765
HepatitisC6	84,5161	0,633	0,089	0,594	0,613	0,807
HepatitisC7	81,2903	0,545	0,122	0,563	0,554	0,782
HepatitisC8	81,2903	0,543	0,130	0,594	0,567	0,796
Post-OperativeC1	60	?	0,705	0,6	?	0,403
Post-OperativeC2	63,333	?	0,646	0,633	?	0,368
Post-OperativeC3	71,111	?	0,711	0,711	?	0,328
Post-OperativeC4	66,666	?	0,681	0,667	?	0,447
Post-OperativeC5	61,111	?	0,608	0,611	?	0,468
Post-OperativeC6	68,888	?	0,719	0,689	?	0,328
Post-OperativeC7	61,111	?	0,701	0,611	?	0,323
Post-OperativeC8	60	?	0,682	0,6	?	0,328
BreastCancerNB	72,7273	0,7808	0,5647	0,8507	0,8142	0,7034
HepatitisNB	82,2258	0,5789	0,1300	0,6875	0,6285	0,8519
Post-OperativeNB	70					
BreastCancerDT	75,43	0,7559	0,7294	0,9552	0,8439	0,5509
HepatitisDT	81,93	0,5769	0,0894	0,4687	0,5172	0,7079

Post-OperativeDT	70					

- **Differences between datasets**

Dataset	NumSamples	ClassType	ClassDistribution	Atributte Characteristics
BreastCancer	286	Binary	201/85	Linear/Nominal
Hepatitis	155	Binary	32/123	Nominal/ Numerical
Post-Operative	70	Multiclass	2/24/64	Nominal/ Numerical

- **Last conclusions**

By way of conclusion, we can highlight the differences that neural networks provide based on their number of layers, learning rate or momentum, these being keys so that the prediction that we want to make is carried out in a reliable and optimal way. In turn, compared to the previous classification algorithms used in this set of dataset, we can say that Naive Bayes is the one that a posteriori with the performance measurements obtained provides us with the best result in the most reliable measurements in Hepatits dataset, although with a minimal difference to configurations of neural networks with two layers like C5. On the other hand if we compare Decision Trees with Neural Networks we can say that the first give us a better overall result, even having certain measures in which not as in accuracy, recall or F-measure in BreastCancer, in the case of Hepatits, we have better results with neural networks, although decision trees model do not differ too much in the results. Finally, I would add that Neural Networks have other options in the modeled parameters with the consequent effect on the results, so saying that one algorithm is better than another is something relative to these.In turn, also in decision trees we have the possibility of applying overfitting techniques such as pruning or early stopping in order to optimize the results, coinciding with the previous criteria applied to neural networks. However, these comparisons give us a global vision of their behavior for possible election decisions.